

Introducción a RIA

Introducción

El objetivo principal de la asignatura es formar al estudiante en el diseño y desarrollo de aplicaciones web ricas.

Los objetivos de la asignatura son:

- Introducir al estudiante en la programación en la plataforma Flex
- Fomentar las habilidades de trabajo en equipo, y la colaboración entre grupos de trabajo
- Entrenar las capacidades de comunicación y presentación oral de los trabajos realizados.

Introducción

Temario

1. Objetivos generales del curso
2. Introducción a las aplicaciones web(Historia, protocolo HTTP)
3. Introducción a la programación del navegador de Internet
4. Plataforma RIA
 - Gráficos vectoriales
 - Manejo de animación
 - Reproducción de contenido multimedia
5. Integración lógica con el servidor
 - Patrones MVC y MVVP
 - Web Services

Introducción

Forma de evaluación

Primera parte del semestre

- Trabajos prácticos y presentación de temas

Segunda parte del semestre

- Implementación de trabajo obligatoria y presentación del mismo

Introducción

¿QUÉ ES UNA APLICACIÓN WEB?

Es una aplicación software que se aloja total o parcialmente en un servidor web y que se codifica en un lenguaje soportado por los navegadores web para ser visualizada y poder interactuar con ella.

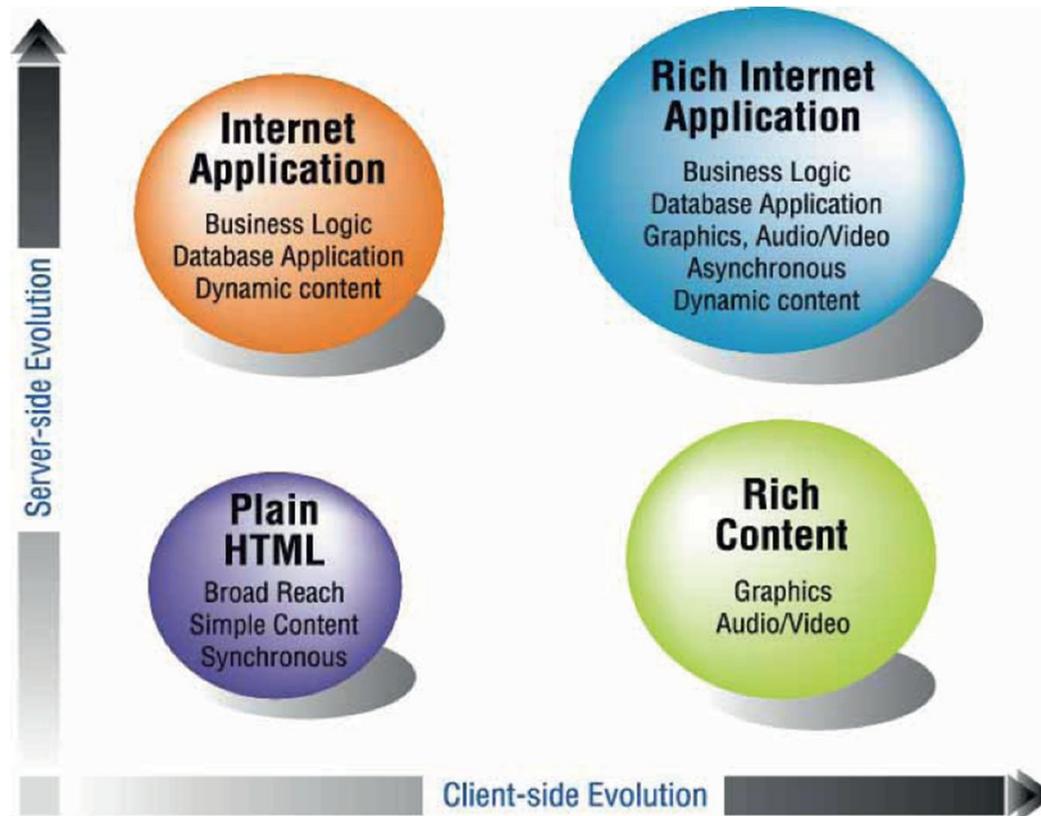
Introducción

Evolución

1. Páginas estáticas
2. CGI (common gateway interface)
3. PHP, ASP
4. Servlets, JSP, ASPX, Ajax
5. RIA

Introducción

Evolución



Introducción

¿Qué es una RIA?

Una RIA (*Rich Internet Application*) es un nuevo tipo de aplicación Web cuyo objetivo es incrementar y mejorar las opciones y capacidades de las aplicaciones Web tradicionales.

Aparecen para subsanar ciertas limitaciones en la capa de presentación de los actuales navegadores Web y del lenguaje HTML.

Permiten, entre otras cosas, mejorar la experiencia del usuario al utilizar la aplicación, la ejecución de contenido multimedia y la carga de aplicaciones online/offline.

Existen distintas tecnologías con las cuales construir aplicaciones RIA.

Introducción

Aplicación Web Vs. RIA

Las aplicaciones web se ejecutan nativamente desde el navegador. Pero existen algunas aplicaciones que funcionan desde el navegador pero además requieren la instalación de un software en el ordenador para poder utilizarse.

Estas aplicaciones se denominan Aplicaciones de Internet Ricas. El motivo de usar este software adicional es que hay muchas funcionalidades que los navegadores no pueden ofrecer, y él enriquece a las aplicaciones web ofreciendo dichas funcionalidades.

Introducción

Características de una RIA

- Mejora de la experiencia visual gracias a la aportación de nuevos componentes más avanzados.
- Permite crear aplicaciones más atractivas mediante la utilización de audio, video y gráficos.
- La mayoría de tecnologías RIA se basan en el lenguaje de programación XML, tanto para las interfaces gráficas como para el intercambio de datos.
- Conectividad. La mayoría de RIAs permiten ejecutar las aplicaciones *online/offline*.

Introducción

Características de una RIA

- Disminuye el ancho de banda utilizado en el uso de la aplicación ya que permite almacenar más información en el cliente de manera que se reduce el uso de transacciones HTTP. Además se minimiza la memoria utilizada en el servidor Web.
- Requiere la utilización de un *cliente rico* en el lado del usuario, que será el motor de la tecnología RIA utilizada (habitualmente será un *plugin*).
- Visualización y ejecución en múltiples plataformas y dispositivos heterogéneos.

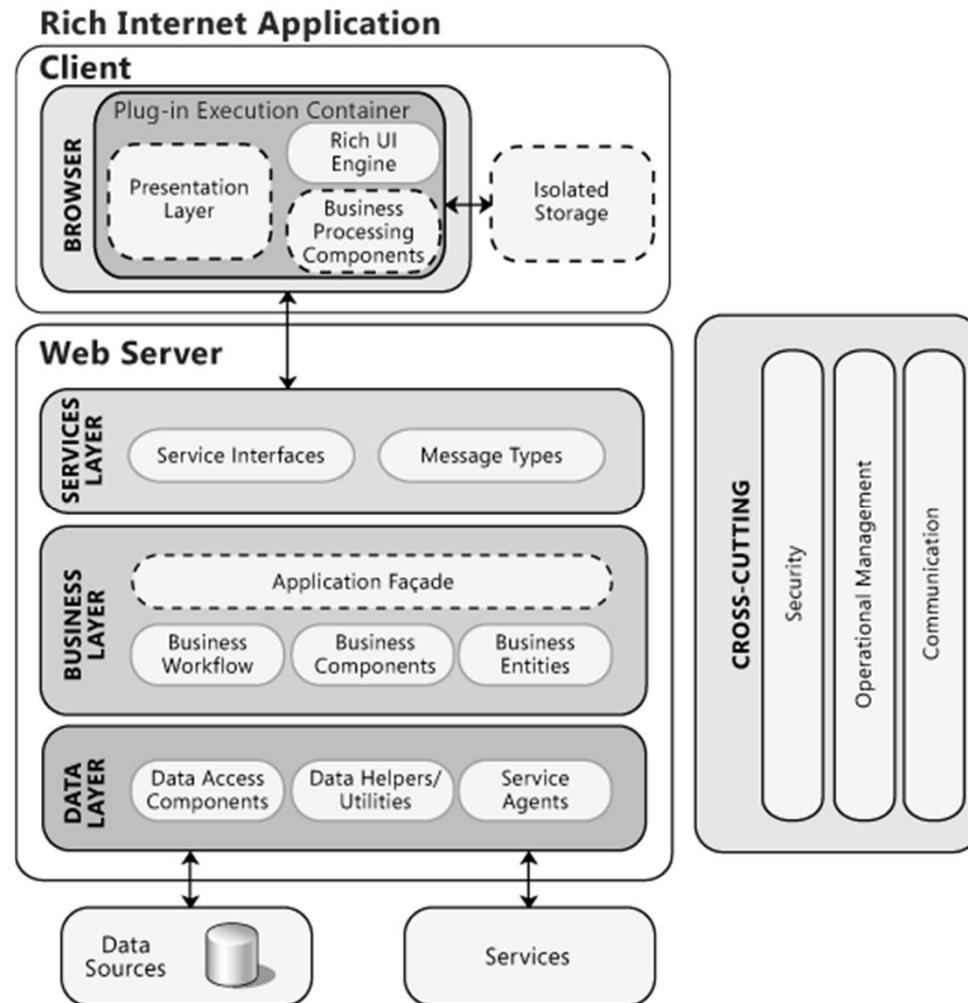
Introducción

Características de una RIA

- Cada interacción del usuario no necesita de una interacción con el servidor
- Los datos pueden ser obtenidos desde el servidor sin necesidad de actualizar toda la pantalla.
- Solo se comunica con el servidor cuando es necesario.
- Facilidad de despliegue
- Facilidad de administración
- Facilidad de desarrollo

Introducción

Estructura de una RIA



Introducción

Tecnologías

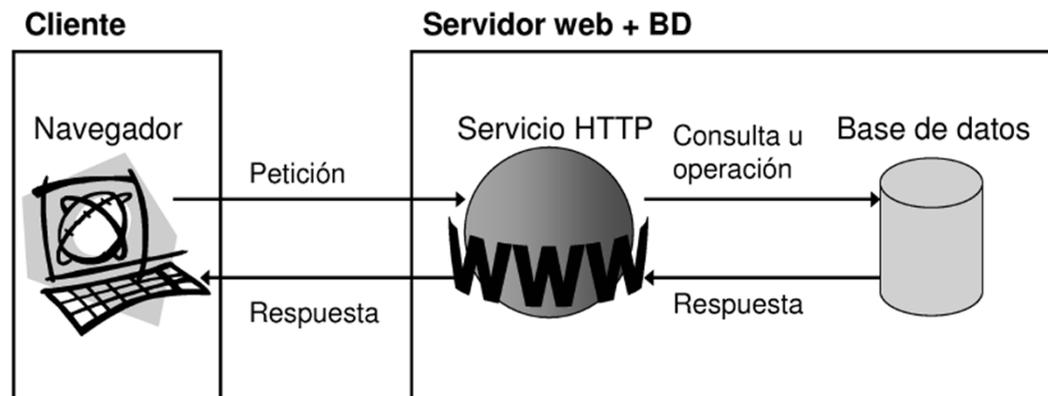
- Adobe Flex
- Adobe Flash
- OpenLazlo
- Silverlight
- Ajax
- HTML5

Introducción

Arquitectura de las aplicaciones Web

Arquitectura en dos capas

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y se ejecuta en una PC y envía solicitudes a un servidor para ejecutar operaciones más complejas.



Introducción

Arquitectura de las aplicaciones Web

Arquitectura de tres capas

La arquitectura de tres capas introduce una capa intermedia en el proceso.

El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de presentar los datos un formato adecuado.

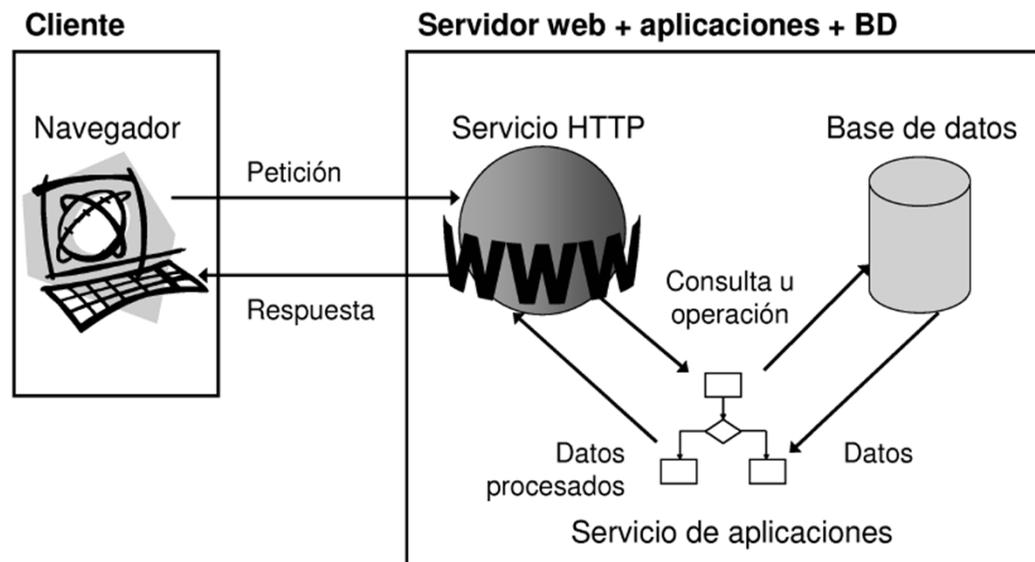
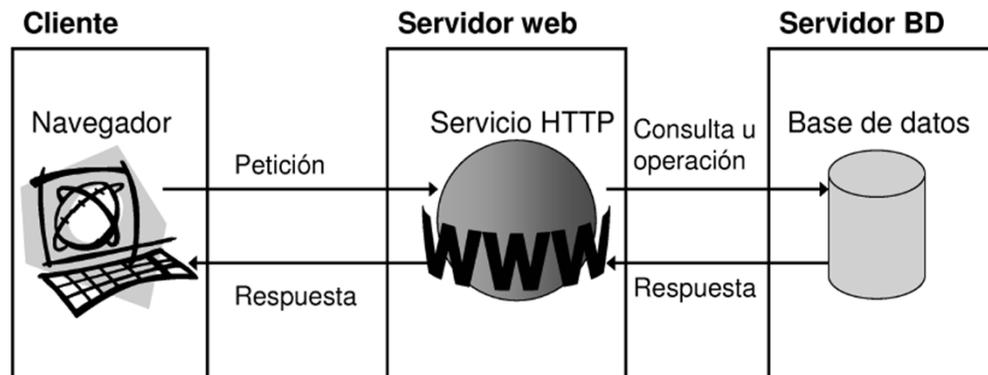
El segundo nivel está referido habitualmente a algún tipo de programa o script.

El tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Introducción

Arquitectura de las aplicaciones Web

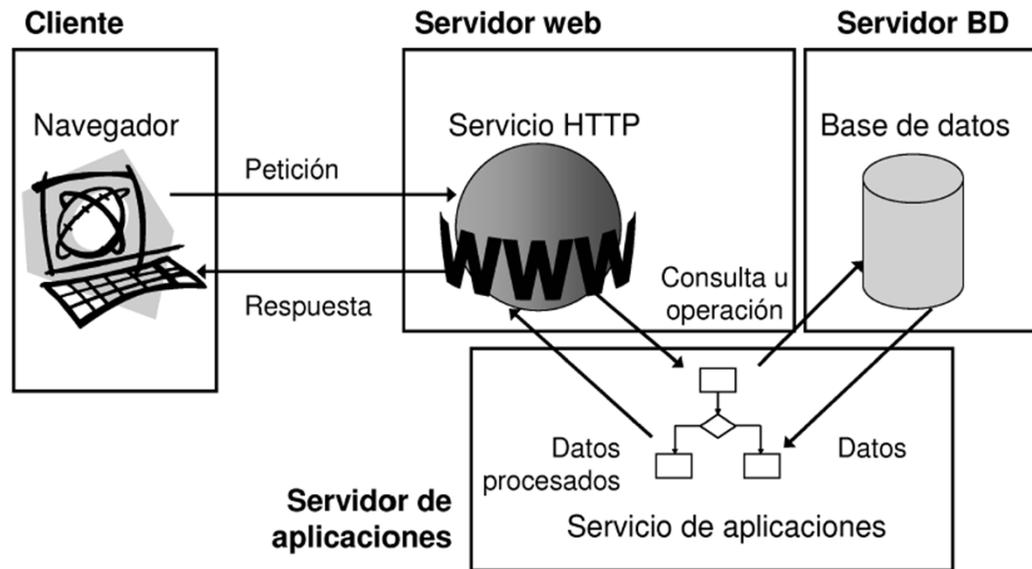
Arquitectura de tres capas



Introducción

Arquitectura de las aplicaciones Web

Arquitectura de tres capas



Introducción

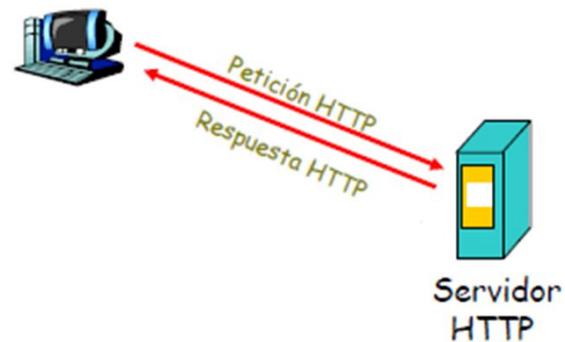
Protocolo HTTP

Los elementos software de la arquitectura web (clientes, servidores, proxies) utilizan el protocolo HTTP para comunicarse.

HTTP define la sintaxis y la semántica que utilizan estos elementos para comunicarse

Permite la transferencia de documentos web

Es un protocolo basado en el esquema petición/respuesta.



Introducción

Protocolo HTTP

- Basado en mensajes. (Texto plano)
Compuestos de una línea inicial, de una cabecera y de un cuerpo.
El mensaje es la unidad fundamental de la comunicación HTTP.
- No maneja estados.
Hay ausencia de estado tras cada par petición-respuesta
Tras la respuesta, el servidor cierra inmediatamente la conexión.
No existe el concepto de sesión.

Introducción

Protocolo HTTP

El cliente envía una petición al servidor en forma de mensaje texto, incluyendo:

- Una línea inicial con el ***método de solicitud, la URL del recurso*** solicitado y la ***versión del protocolo***.
- Una lista de campos, consistente en modificadores de la petición, información del cliente, etc.
- Un posible cuerpo de contenido.

El servidor responde con un mensaje donde se incluye:

- Una línea de ***estado, con la versión del protocolo y un código*** de resultado.
- Una lista de campos, donde se incluyen entre otras cosas: el ***tipo MIME de la respuesta, información del servidor, entidades*** de meta-información, etc.
- Un cuerpo con el contenido del recurso solicitado (opcional).

Introducción

Protocolo HTTP

Métodos HTTP

GET: Solicita un documento al servidor.

Se pueden enviar datos en la URL

HEAD: Similar a GET, pero sólo pide las cabeceras HTTP.

POST: Manda datos al servidor para su procesado.

Similar a GET, pero además envía datos en el cuerpo del mensaje.

PUT: Almacena el documento enviado en el cuerpo del mensaje.

DELETE: Elimina el documento referenciado en la URL.

TRACE: Rastrea los intermediarios por los que pasa la petición.

OPTIONS: Averigua los métodos que soporta el servidor.

En una caché sólo se guardan las respuestas de las peticiones realizadas con GET y HEAD (POST no).

Introducción

Protocolo HTTP

Códigos de error

1xx: Mensaje informativo.

2xx: Éxito

200 OK

201 Created

202 Accepted

204 No Content

3xx: Redirección

300 Multiple Choice

301 Moved Permanently

302 Found

304 Not Modified

4xx: Error del cliente

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

5xx: Error del servidor

500 Internal Server

Error

501 Not Implemented

502 Bad Gateway

503 Service Unavailable

Introducción

HTML

Es un lenguaje de marcas, es interpretado por los navegadores y basa su sintaxis en elementos llamados etiquetas.

A través de las etiquetas se van definiendo los elementos del documento, como enlaces, párrafos, imágenes, etc.

Un documento HTML estará constituido por texto y un conjunto de etiquetas para definir la forma con la que se tendrá que presentar el texto y otros elementos en la página.

La etiqueta presenta frecuentemente dos partes:

Una apertura de forma general `<etiqueta>`

Un cierre de tipo `</etiqueta>`

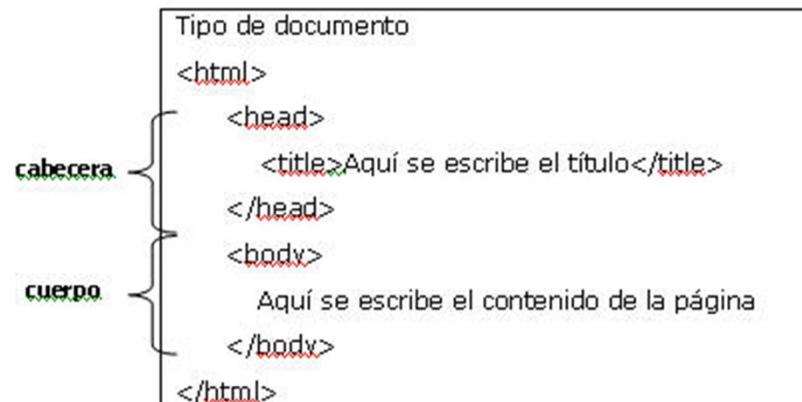
Todo lo incluido en el interior de esa etiqueta sufrirá las modificaciones causadas por la característica de la etiqueta.

Introducción

HTML

Partes de un documento HTML

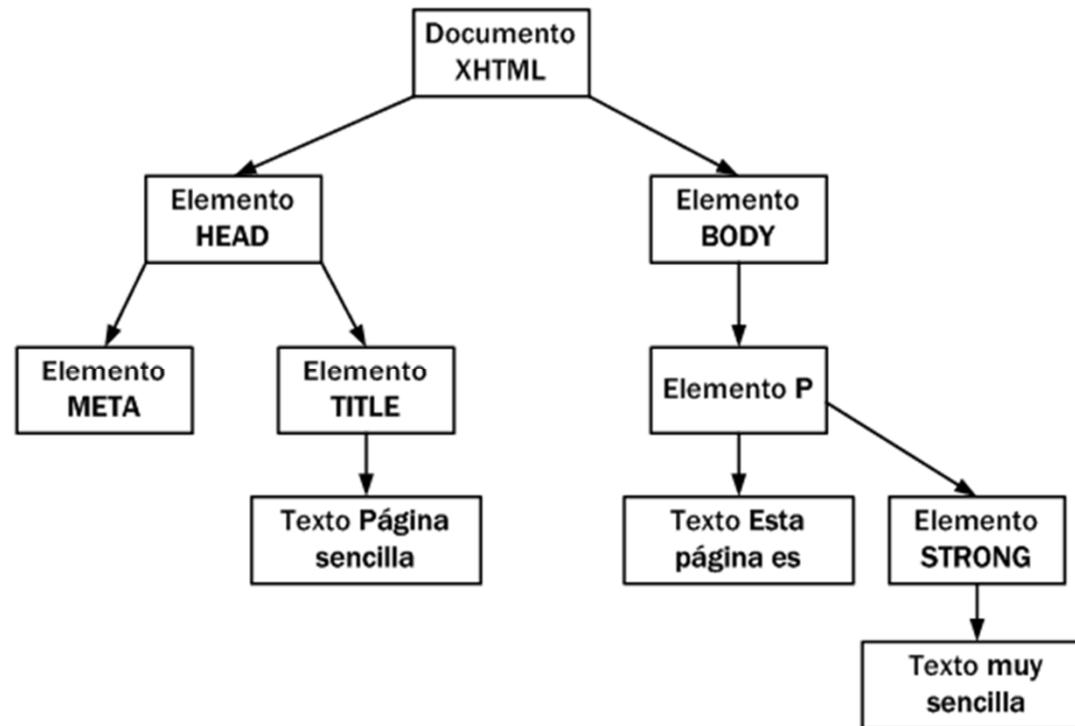
- Un documento HTML está delimitado por la etiqueta `<html>` y `</html>`.
- Un encabezado, delimitado por `<head>` y `</head>` donde se colocan etiquetas de índole informativo como por ejemplo el título.
- El cuerpo delimitado por `<body>` y `</body>`, que será donde se coloca el contenido (texto imágenes) delimitados a su vez por otras etiquetas .



Introducción

HTML

Partes de un documento HTML



Introducción

CSS

Una hoja de estilo es un conjunto de especificaciones que declaran como deben mostrarse los componentes de una página web, es decir, cómo debe interpretar el navegador, a efectos de presentación visual, los distintos elementos presentes en una página web. El formato de cada elemento de una página web se especifica mediante declaraciones de estilo denominadas reglas.

Introducción

CSS

Reglas

Una regla CSS se compone de tres partes:

1. Selector
2. Propiedad
3. Valor

Una regla simple organiza los tres componentes anteriores de acuerdo a la siguiente estructura:

```
selector {propiedad: valor;}
```

Una regla puede contener más de una propiedad con sus respectivos valores, separadas entre ellas por punto y coma (se considera una buena práctica añadir también punto y coma al final).

Introducción

CSS

Declaración

Las reglas de las hojas de estilo se pueden declarar en tres sitios distintos:

1. En la página web `<style type="text/css">`
2. En un archivo externo `<link rel="stylesheet" href="estilo.css" type="text/css" />`
3. En un elemento `<p><acronym style="font-style: italic`

Introducción

CSS

Cajas

Desde el punto de vista de CSS, todo lo que hay en una página web es una caja. Lo que hace un selector es identificar la caja a la que se aplica el formato.

Las cajas tienen:

Contenido (*content*). El contenido de la caja: texto, imagen, etc.

Relleno (*padding*). La distancia entre el contenido y el borde.

Borde (*border*). El borde de la caja.

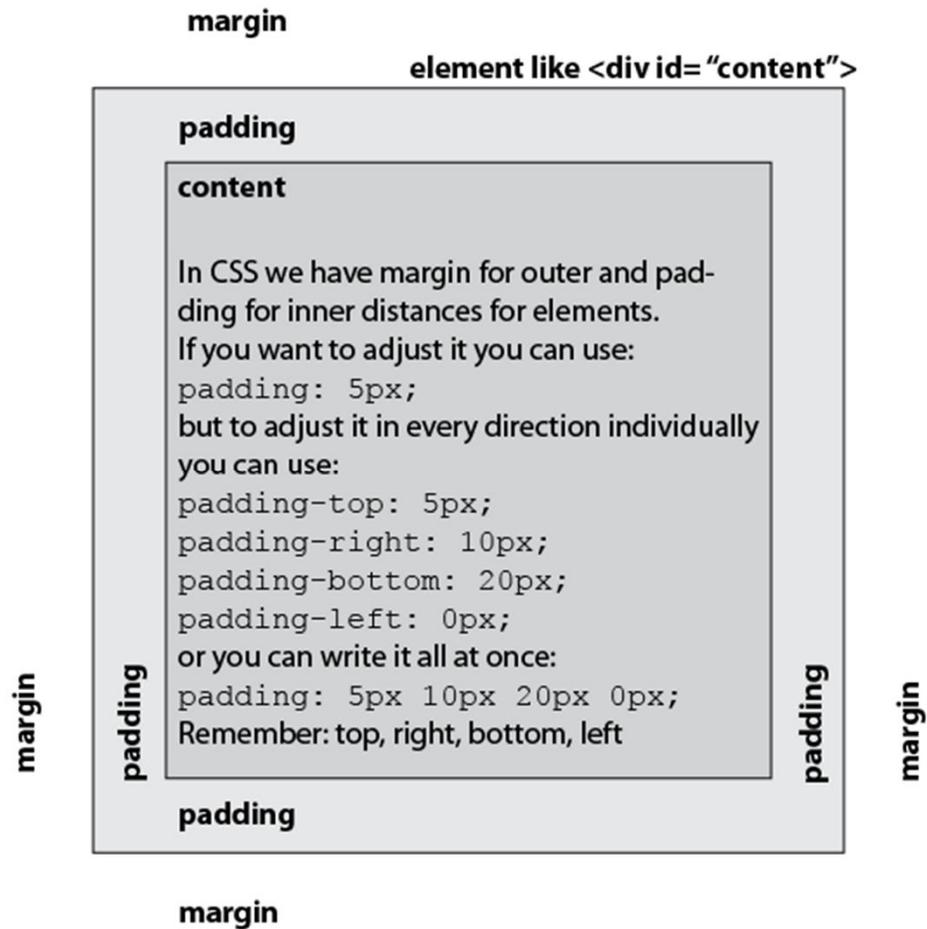
Margen (*margin*). La distancia entre el borde y el elemento contenedor u otro elemento adyacente.

Las hojas de estilo pueden aplicar formato (color, anchura, tipo de línea, etc.) a cualquiera de los componentes de esta caja, así como permite posicionar estas cajas en el conjunto de la página.

Introducción

CSS

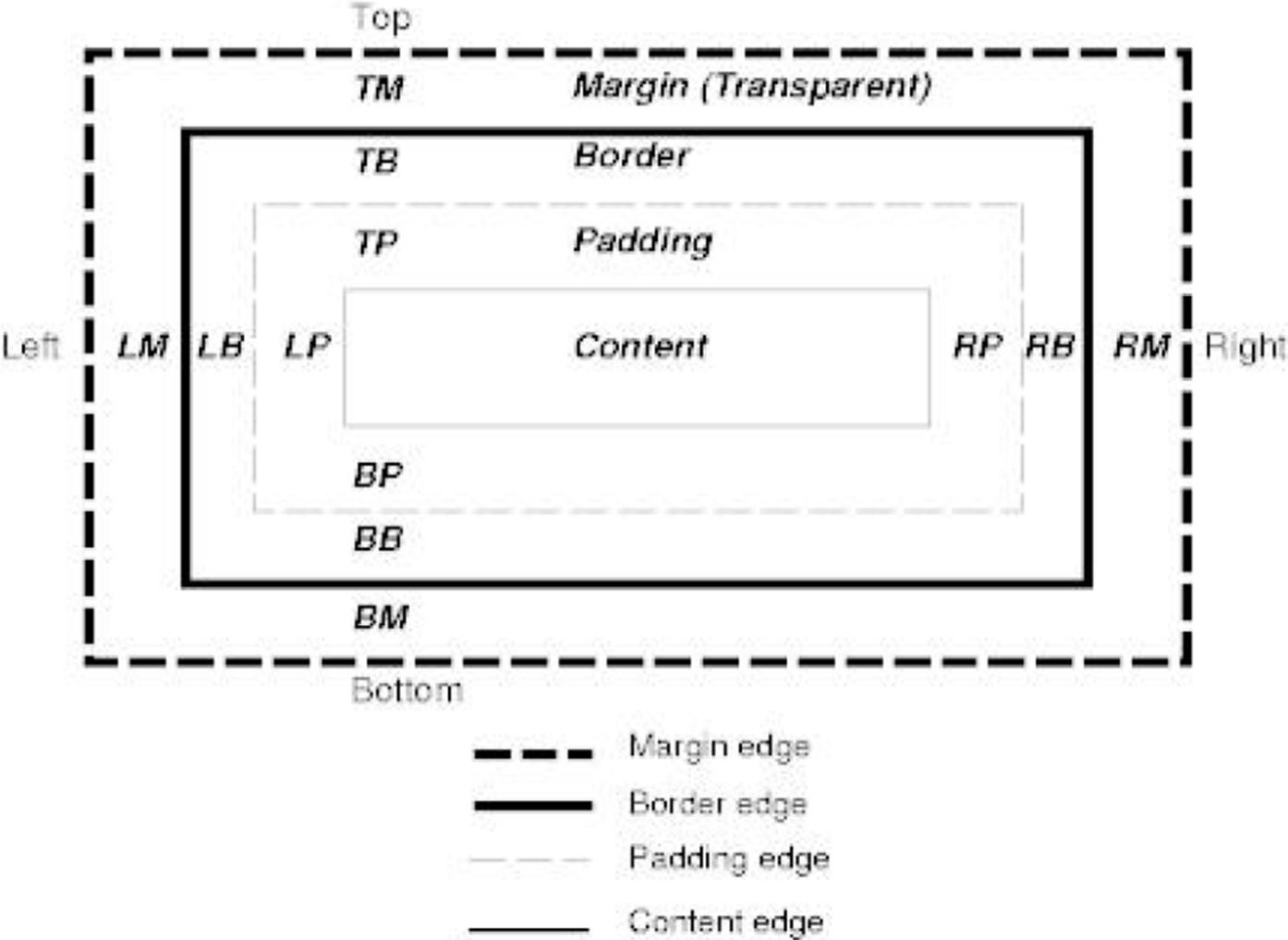
Cajas



Introducción

CSS

Cajas

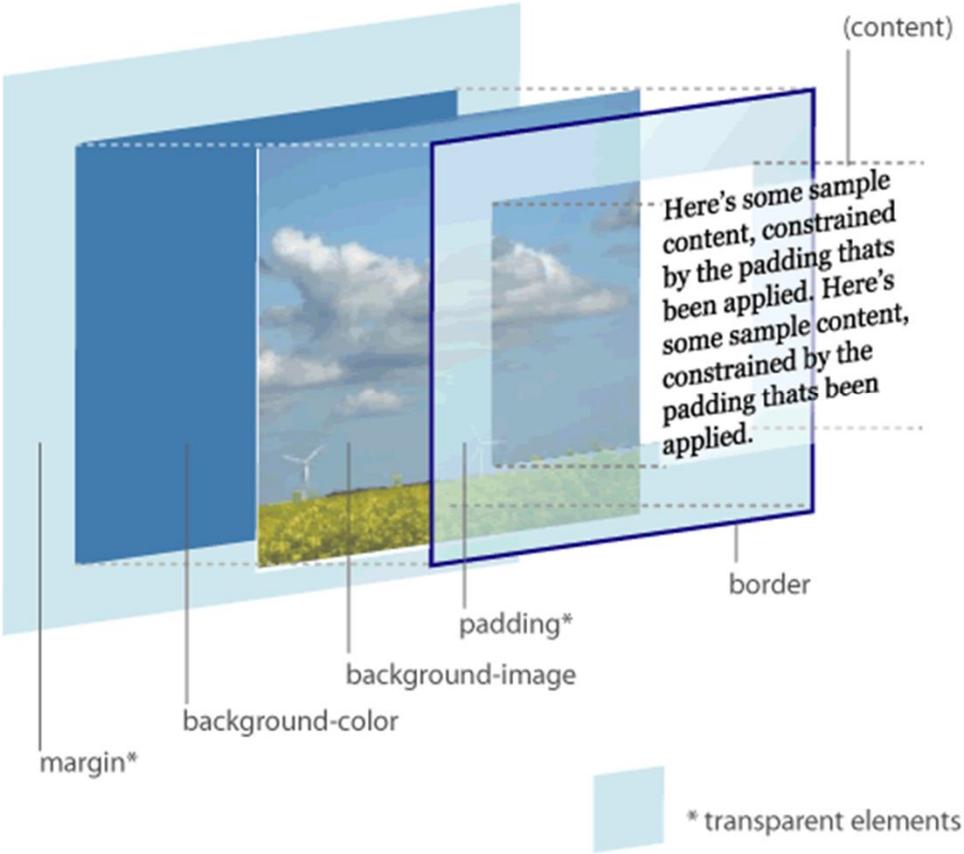


Introducción

CSS

Cajas

THE CSS BOX MODEL HIERARCHY



Introducción

CSS

Herencia

Las cajas están contenidas dentro de otras cajas y heredan los estilos de la caja contenedora si no hay una regla de nivel específico.

Por ejemplo, si declaramos un tipo de letra para el elemento body, todos los elementos que están dentro, como h1, h2, p, etc. heredarán este tipo de letra.

Introducción

CSS

Tipos de selectores

Existen dos grandes categorías de selectores:

- Elementos: son los nombres correspondientes a elementos del lenguaje (X)HTML como body, h1, p, table, etc.
- Nombres propios: son nombres que puede crear el autor de la hoja de estilo. De esta clase de selectores, existen, a su vez dos tipo: de clase (class) y de identidad (id).

Introducción

CSS

Selectores de clase

Estos selectores se declaran mediante una palabra propia que asigna el autor de la hoja de estilo. Esta palabra va precedida por un punto.

Selectores de elemento

Se declaran utilizando como selector un elemento (X)HTML, p.e.:

```
h1 { color: blue }
```

De este modo, el aspecto de todos y cada uno de los elementos que forman el lenguaje (X)HTML puede ser modificado mediante un regla.

Introducción

CSS

Selectores de identidad

Los selectores de identidad se nombran mediante una palabra propia precedida por el símbolo #. Por ejemplo:

```
#nav { background-color: red; }
```

Los selectores de identidad solo se pueden aplicar a un elemento en cada página. Dicho al revés, y tomando el ejemplo anterior: en cada página solamente puede haber un elemento #nav

Introducción

CSS

Uso de div y span

El lenguaje (X)HTML dispone de dos elementos con gran potencialidad cuando se utilizan junto con las hojas de estilo, y cuya característica principal, es que no poseen ningún significado intrínseco.

Estos son:

Div

span

El elemento div es de bloque y el elemento span es de línea.

Los elementos de bloque tienen un salto de línea integrado.

(<p>). Un elemento de línea, por el contrario no

genera ningún espacio a su alrededor, sino que se mantienen en la misma línea que lo contiene. Además de utilizarlos para cuestiones de apariencia, contribuyen a añadir valor semántico.

Introducción

CSS

Posicionamiento

Las hojas de estilo proporcionan propiedades que permiten posicionar de forma fija, absoluta o relativa (*fixed, absolute, relative*) las cajas en una página web.

Introducción

XHTML

Es HTML con sintaxis XML.

Conserva la mayoría de los elementos y atributos de HTML.

La más destacable diferencia es la ausencia de elementos y atributos relacionados con estilo (tipos de letra, colores, etc.)

Introducción

XHTML

- Refuerza la separación entre contenido y presentación, eliminando de la especificación aquellos elementos y atributos relacionados con el estilo.
- Existen reglas estrictas acerca del formato que debe tener un documento.
- Modularización, existen distintas versiones de XHTML permiten la extensión de XHTML sin rehacer de nuevo la especificación, añadiendo nuevos módulos aprovechando la capacidad de extensibilidad de XML.

Introducción

XHTML

Versiones de XHTML

XHTML 1.0 con tres variantes:

Transitional: permite el uso de las capacidades de presentación de HTML y está pensado para trabajar con navegadores con soporte de CSS limitado.

Strict: no se permite el uso de los elementos y atributos relacionados con aspectos de presentación. Pensado para ser usado con hojas de estilo CSS.

Frameset: permite el uso de marcos (frames) para dividir la ventana del navegador.

XHTML 1.1 versión, que parte de XHTML 1.0 strict. No permite el uso de elementos y atributos relacionados con el estilo.

Se introduce el concepto de modularización.

XHTML Basic 1.0 y 1.1] versiones simplificadas de XHTML 1.1 pensadas para dispositivos de capacidades más limitadas de procesamiento y presentación.

Introducción

XHTML

Diferencias entre HTML y XHTML

- Los nombres de elementos y atributos deben escribirse en minúscula .
- Los valores de los atributos deben escribirse entre comillas.
todos los elementos tienen marca de inicio o finalización. Por ejemplo: `
`, `<p></p>`.
- La anidación de elementos debe ser correcta.
- Los elementos y atributos utilizados para especificar preferencias en cuanto al estilo de la información (como por ejemplo `` o el atributo `bgcolor`) desaparecen.
- Las características de presentación se establecen ahora mediante el uso CSS.

Introducción

XHTML

Diferencias entre HTML 4 y XHTML

El atributo id sustituye al atributo name en: a, applet, frame, iframe, img y map.

Es obligatorio añadir al comienzo del documento XHTML una declaración DOCTYPE que referencia al DTD (Declaración de Tipo de Documento) donde se indican las reglas de construcción del documento

Introducción

Ejercicios

Hacer un menú utilizando HTML y CSS

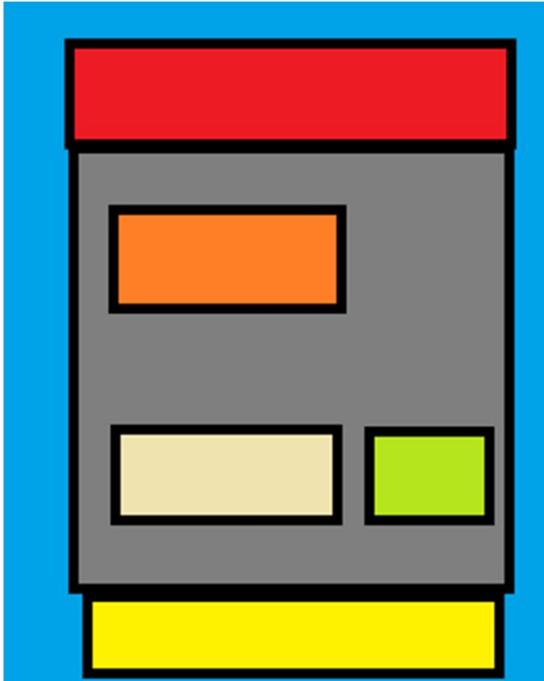


Introducción

Ejercicios

Realizar una maquetación con Html y CSS similar a la figura de abajo

```
.bloque1 { margin-left:100px;width:600px;}  
  
body {background-color: #0099ff;}  
  
#div1 {background-color: #ff0000;}  
#div2 {background-color: grey; height:200px;}  
#div3 {background-color: yellow; height:100px;border-style:solid;  
border-width:5px;}  
#div2-1 {background-color: #ff9933;height:50px;width:50px;margin-left:50px;margin-bottom:50px; }  
#div2-2 {background-color: #ffffcc;float:left; height:50px;width:50px;margin-left:50px; }  
#div2-3 {background-color: #66ff99;height:50px;width:50px; float:left;margin-left:20px;}
```



```
<head>  
  
<title>Ej1</title>  
<link rel="stylesheet" href="./ej1.css" type="text/css" >  
</head>  
  
<body>  
<div id="div1" class="bloque1">  
bloque1  
</div>  
  
<div id="div2" class="bloque1">  
bloque2  
<div id="div2-1" >  
interno 2-1  
</div>  
<div id="div2-2" >  
interno 2-2  
</div>  
<div id="div2-3" >  
interno 2-3  
</div>  
</div><!-- fin div2-->  
  
<div id="div3" class="bloque1">  
bloque3  
</div>  
</body>  
  
</html>
```

Introducción

Ejercicios

Crear un formulario HTML que invoque a un script php . Probar con me ´todo post y con método get.

```
<?php  
$var = $_GET['name'];  
echo $var;  
?>
```

```
<?php  
$var = $_POST['name'];  
echo $var;  
?>
```

Introducción

Ejercicios

- Leer más de HTML y CSS en <http://www.w3schools.com>
- Estudiar Pseudo selectores y pseudo clases

Introducción

Referencias

<http://msdn.microsoft.com/en-us/library/ee658083.aspx>

Bibliografía

Programacion de aplicaciones web-Sergio Lujan

Introducción

Fin